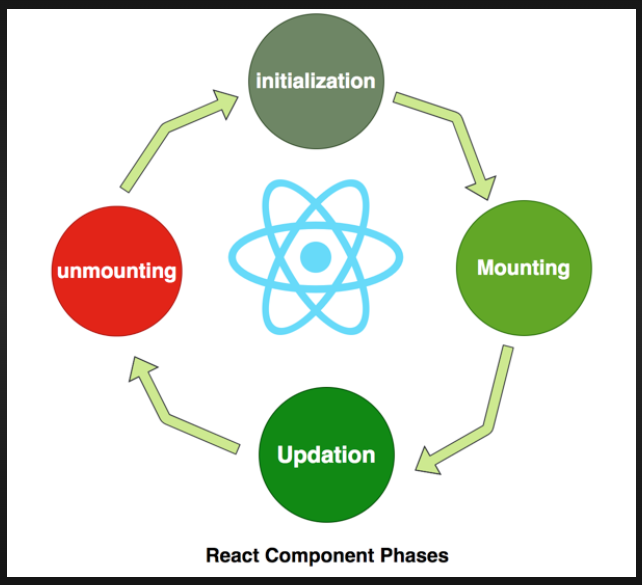
**List & Hooks**

**● Explain Life cycle in Class Component and functional component with Hooks.**

**A.**

**Life cycle:**

****

**(1) Mounting:**

The mounting phase occurs when a component is created and inserted into the DOM.

**(2) Updating:**

The updating phase occurs when a component’s props or state changes.

**(3) Unmounting**

The unmounting phase occurs when a component is removed from the DOM.

**Example:**

import React from "react";

import ReactDOM from "react-dom/client";

class Test extends React.Component {

    constructor(props) {

        super(props);

        this.state = { hello: "World!" };

    }

    componentDidMount() {

        console.log("componentDidMount()");

    }

    changeState() {

        this.setState({ hello: "Tops!" });

    }

    render() {

        return (

            <div>

                <h1>

                    Tech

                    {this.state.hello}

                </h1>

                <h2>

                    <a

                        onClick={this.changeState.bind(

                            this

                        )}

                    >

                        Press Here!

                    </a>

                </h2>

            </div>

        );

    }

    shouldComponentUpdate(nextProps, nextState) {

        console.log("shouldComponentUpdate()");

        return true;

    }

    componentDidUpdate() {

        console.log("componentDidUpdate()");

    }

}

const root = ReactDOM.createRoot(

    document.getElementById("root")

);

root.render(<Test />);

**◙ Functional Component ◙**

Functional Components are some of the more common components that will come across while working in React. These are simply JavaScript functions. We can create a functional components to React by writing a JavaScript function.

**Example of functional components:**

import React,{useState} from "react";

const FunctionalComponents = ()=>{

    const [count, setCount] = useState(0);

    const increase = () => {

        setCount(count + 1);

    }

    return(

        <div style={{margin:'50px'}}>

            <h3> counter app using Functional Components:</h3>

            <h2>{count}</h2>

            <button onClick={increase}>Add</button>

        </div>

    )

}

export default FunctionalComponents;

**◙ Class Component ◙**

This is the bread and butter of most modern web aps built in ReactJS. These components are simple classes (made up of multiple functions that add functionality to the application).

**Example of class components:**

import React,{Component} from "react";

class ClassComponents extends React.Component{

    constructor(){

        super();

        this.state = {

            count: 0

        };

        this.increase = this.increase.bind(this);

    }

    increase(){

        this.setState({count: this.state.count + 1});

    }

    render(){

        return(

            <div style={{margin: '50px'}}>

            <h3> counter app using Functional Components:</h3>

            <h2>{this.state.count}</h2>

            <button onClick={this.increase}>Add</button>

            </div>

        )

    }

}

export default ClassComponents;